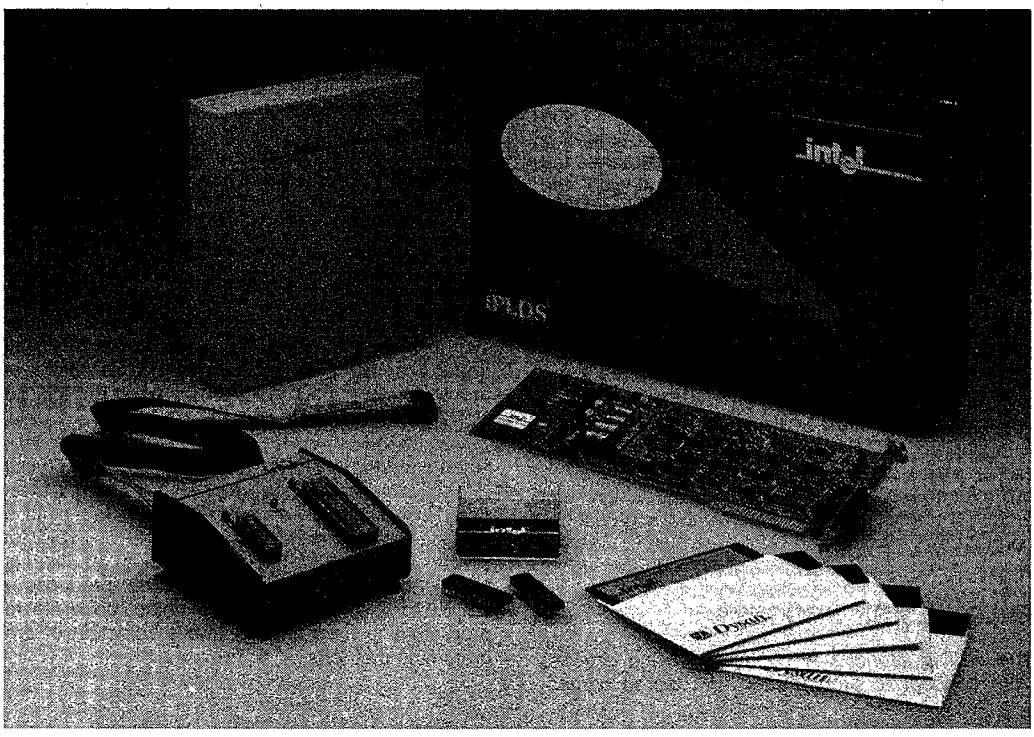
iPLDS
INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM

- **Provides the Necessary Hardware and Software Tools to Quickly Turn Design Concepts Into Programmed Erasable Programmable Logic Devices (EPLD)**
- **Includes Comprehensive, Menu-driven Software with Soft Key Input and On-line Help Messages**
- **A Variety of Programmable Options Available to Program, Read, and Verify EPLD Devices**
- **Includes a Logic Optimizing Compiler That Automatically Minimizes Logic, and Produces the Best Design Fit for the Device Selected**
- **Supports a Variety of Input Methods:**
  - **— Schematic Capture (Optional)**
  - **— Interactive Logic Builder**
  - **— Boolean Equations**
  - **— State Machine Entry (Optional)**
  - **— Design Files Via a Text Editor**
- **Generates Output in the Standard JEDEC File Format**
- **Interfaces with the IBM* PC, PC XT, PC AT, and True Compatibles**
- **Programming Tools to Obtain the Most Utilization of EPLD Resources**
- **Includes Sample Device**

OVERVIEW: The Intel Programmable Logic Development System (iPLDS) provides a powerful set of EPLD development tools. It is an easy to use hardware and software system for creating a logic design, optimizing and custom-fitting the design to a particular EPLD device, and then programming and verifying the EPLD device.

*IBM Personal Computer is a registered trademark of International Business Machines Corporation.



280168-1

October 1986
Order Number: 280168-003

FUNCTIONAL DESCRIPTION

The iPLDS simplifies using EPLD devices in circuit designs. The iPLDS provides all of the software, programming hardware, and documentation needed to convert a designer's hardware logic concept into a fully optimized, tested, and documented device. The designer accomplishes the entire process at his/her desk. The iPLDS interfaces with and runs on an IBM PC or true compatible.

The key to the ease of this process is the comprehensive set of high-level software tools, derived extensively from the techniques used in higher cost CAE workstations and software development processes. The system's software includes a wide choice of design input types, enabling designers to create and implement designs using the user interface matching their application.

As with most other programmable logic software systems, the designer can specify, test, and modify designs with advanced forms of Boolean equations. In addition, the iPLDS supports input from two powerful optional schematic capture packages (PC-CAPS* from P-CAD, and DASH-2* from Future-Net), input using the logic builder program (an easy-to-use interactive netlist entry package), optional state machine entry, and creation of an Advanced Design File (ADF) directly using the text editor, such as Intel's AEDIT text editor.

Advances have been made in each stage of the EPLD design cycle. The software compiles and optimizes the logic design, automatically determines the best way to fit the design into the EPLD device, and graphically displays the programmed device at the individual EPROM bit level. The software also programs, reads, and verifies the EPLD device using the system's programming hardware.

The software is designed for ease of use. The entire software package is comprised of nested menus. The bottom of the screen displays helpful messages, suggesting what to do. A separate help function is always available when a further explanation of a function is needed. Errors are identified with descriptive messages. The process is further simplified by the use of interactive graphics during design input and while viewing the design fit using the gate interconnect preview function.

EPLD DESIGN PROCESS

The iPLDS supports a complete design process from concept to programmed and tested components. The Intel Programmable Logic Software (iPLS) controls the entire process (refer to Figure 1).

Design Input

The logic design can be entered using any of the following methods:

Logic Builder: The logic design can easily be entered using the logic builder program, which uses a combination of questions and pictures to prompt the designer for inputs and outputs of logic elements. The program guides the designer through the entire design entry process by prompting for necessary information and showing a screen display, one device at a time, with input signals on the left side, and output signals on the right (refer to Figure 2).

The design entry process starts with an output pin of the EPLD device. A device to drive the output pin is selected from a menu of available logic primitives. Then the system prompts the designer for the node names of each input to the primitive device. Primitive devices to drive each input node are then selected, and so on, until the entire logic circuit has been created. The circuit can be edited during initial entry and also when down-loaded from disk storage or an EPLD device. Comments can also be added. The circuit is corrected-by-design from the start, as the logic builder detects and identifies violations to basic design rules. If the design can be better described using a Boolean equation or a state machine specification, these can be directly entered into the circuit, using built-in entry functions. An Advanced Design File (ADF) is automatically created when the design input is complete.

State Machines (optional): State machine designs can be entered using the optional iSTATE software. iSTATE provides considerable flexibility in ways to specify state machine designs, supporting multiple syntaxes for state definition, specification of state transitions, inputs, and outputs and provisions for intermixing state machine and Boolean equation designs.

Once a state machine design has been coded using iSTATE, the program is input to the iPLS software which optimizes the logic, determines a best fit of the state machine design to the EPLD selected (even including automatic selection of flip-flop types), compiles the program, and produces a JEDEC or Intel hex formatted object code file for programming using one of Intel's selection of programming methods.

*PC-CAPS is a registered trademark of P-CAD Corporation.

*DASH-2 is a registered trademark of FutureNet Corporation.

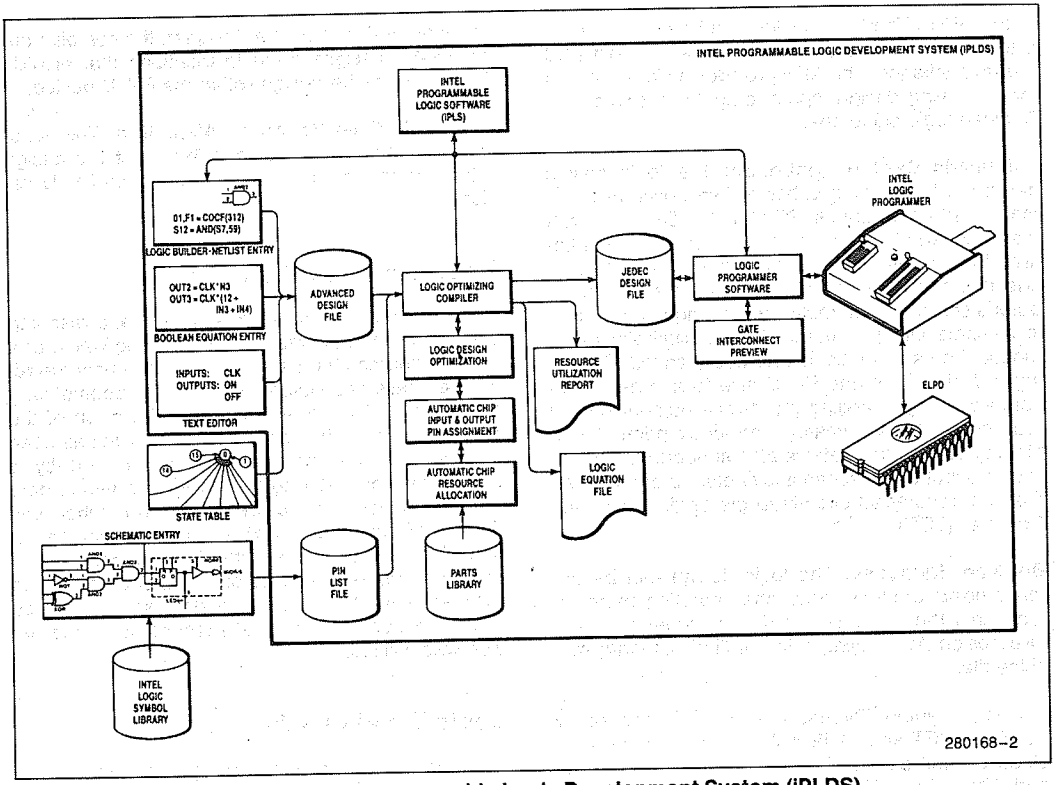


Figure 1. Intel Programmable Logic Development System (iPLDS)

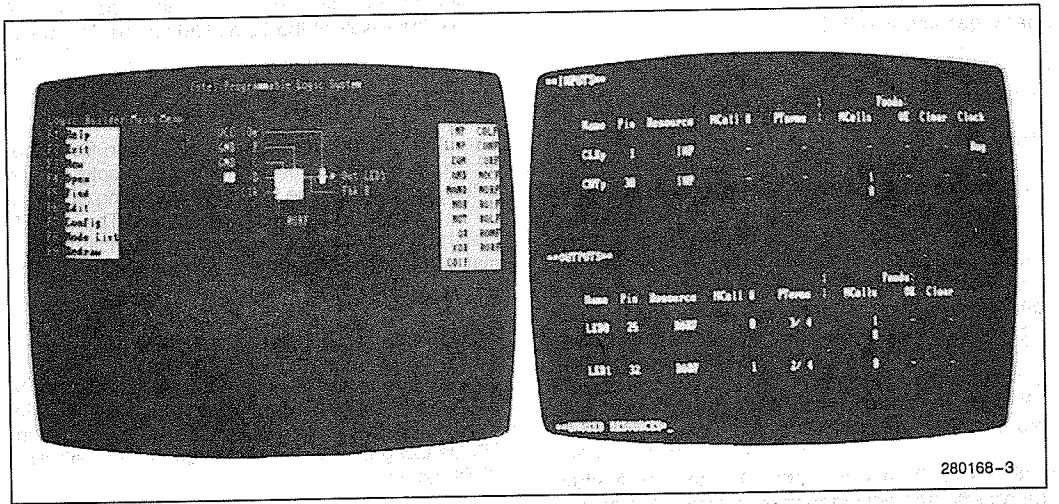


Figure 2. Logic Builder and Utilization Screen Samples

Text Editor: The logic design can be entered using a text editor to create an ADF similar to most PLD design packages. The ADF provides a simple format for specifying design inputs, outputs, net lists, and Boolean logic equations.

Schematic Capture (optional): The logic design can be entered using either of two powerful schematic capture programs, PC-CAPS or DASH-2. With these schematic entry programs, specially configured to work with Intel's logic symbol libraries, logic design schematics can be drawn on the host computer's screen by interactively entering primitive logic symbols using a menu of logic symbols and a mouse. The schematic can be easily edited. Changing a design from one EPLD size to another is accomplished by changing the device number on the drawing of the schematic. Hard-copy printout and plotting of the schematic is also supported. The pin list file ouput of the schematic capture programs can then be transferred directly to the Logic Optimizing Compiler (LOC).

Boolean Equations: The logic design can be entered using Boolean equations, inserting them directly into the Logic Builder design, schematic capture design ADF (using a text editor), or state machine file.

The logic symbol libraries are available separately as iSLIBFNET and iSLIBPCAD. These products include the full device libraries to support both Intel and Altera Corp. EPLD devices, and all necessary software interfacing between the schematic capture packages and the iPLDS.

Logic Optimizing Compiler (LOC)

The LOC accepts the design input from an ADF or a pin list file. Once the logic design is accepted, the LOC begins to compile the input code in three stages.

Logic Design Optimization: The LOC converts the input file to Boolean equations. At this time, logical and syntactical error checks are made. The Boolean equations are then combined into an expanded sum-of-products form. The LOC then performs heuristically selected optimization algorithms (including De Morgan's theorem) to reduce the design to the minimum number of terms.

Automatic Chip Input and Output Pin Assignment: After the optimization is complete, the compiler automatically fits the design into the specified EPLD device using device parameters read from the software parts library. If input and output pins have not been assigned by the designer, the compiler automatically assigns them to locations that provide the best fit of the design within the EPLD device.

Automatic Chip Resource Allocation: This function determines the best possible fit of the design within the format structure of the chosen EPLD device.

Resource Utilization Report

Once the logic optimization is complete, a resource utilization report is created documenting which of a part's resources have been utilized and how the resources have been used. The report is automatically stored as a disk file. A statement at the top of the report indicates whether or not the design has been successfully implemented. This is followed by a header section detailing the designer's name, date the design was entered, EPLD device number, and the title of the design. Next is a pictorial representation of the EPLD device, with all pins labeled. Then details of the input and output pins, and any buried registers are listed. The report also lists any unused device resources, and what percentage of the device was utilized.

Logic Equation File

At the completion of the logic optimization, a Logic Equation File (LEF) is also created. The LEF is a version of the ADF with all logic minimized. The LEF shows the result of the Logic Optimizing Compiler.

JEDEC Design File

The LOC also produces a JEDEC design file of object code, which can be programmed directly into an EPLD device using the Logic Programming Software (LPS) and the Intel logic programmer. The JEDEC (Joint Electron Device Engineering Council) file format is a standard data transfer format.

JEDEC to HEX Conversion

The JEDEC file can be converted to an Intel Hex File Format using a simple conversion program. The Intel Hex File Format code can then be used to program EPLDs using Intel Universal Programmers with iUP-GUPI modules.

Logic Programmer Software (LPS)

The LPS controls the programming, reading, and verifying of the EPLD device by the Intel logic programmer. The gate interconnect preview feature of

the LPS provides a windowed view into the structure of the EPLD device, graphically displaying how the design was implemented into the device (refer to Figure 3). The feature enables the designer to get a complete view of the EPLD device, showing the status of individual EPROM bits. The actual bit pattern and I/O drivers can be checked, and individual bits may be altered. This feature can be used before the EPLD device has been programmed, or after reading a previously programmed EPLD device.

Intel Logic Programmer

Programming the EPLD is accomplished by use of the Intel logic programmer, which consists of an interface card (installed in an IBM PC, or true compatible) and a separate programmer (pod) that is connected to the card by a ribbon cable. The Intel logic programmer uses a fast programming algorithm to program most designs in less than one minute.

While programming, the Intel logic programmer also performs a double verification of the bit pattern. The programmer verifies each bit after it is programmed, and it verifies the bit pattern again after the entire EPLD device is programmed.

The Intel logic programmer also programs the EPLD security and turbo bits. The security bit, once programmed, prevents programming, reading, and verifying of the device. The turbo bit, once programmed, prevents the device from going into a stand-by mode. Although the device will consume more power in turbo mode, the propagation delay through the device is reduced. To be reprogrammed, the device must be erased with ultraviolet light.

NOTE:
The iPLDS includes a programmer pod for the 300 and 1200 gate equivalent devices. Other logic programmers are available for programming devices with different pin counts and package styles.

iUP-GUPI

The iUP-GUPI is a generic module that enables the iUP-200A/201A Universal Programmer and the Intel Personal Development System (iPDS™) to accept low-cost plug in adaptors that configure the system to support a wide variety of programmable devices (including EPLDs). Table 1 lists the EPLD devices supported by the iPLDS system, and the Intel Logic Programmer pods and GUPI adaptors that will program them.

Table 1. Intel Programmer Logic Development System Programming Support

Device	Equivalent Gate Count	Intel Logic Programmer Pod	iUP-GUPI Adaptor
5C031	300	included in iPLDS	GUPI LOGIC-12
5C060	600	iLP900	GUPI LOGIC-9
5C090	900	iLP900	GUPI LOGIC-9
5C121	1200	included in iPLDS	GUPI LOGIC-12
5C180	1800	iLP1800	GUPI LOGIC-18

NOTE:
Intel Programmers also support programming of equivalent Altera Corp. second-source parts.

iUP-GUPI and GUPI Logic Adaptors

The iUP-GUPI and assorted GUPI LOGIC adaptors provides an alternative programming solution for Intel's H-series and Altera EPLD devices, when purchased with the iPLS, Intel's Programmable Logic Software. This complete set of software is available separately (i.e., without the iLP programmer pod and IBM interface card).

By selecting a system consisting of the iPLS software, iUP-201A (with iPPS software for the IBM PC, PC XT, or PC AT), and iUP-GUPI, no expansion slots are used in your PC (since the iUP communicates via the PC's RS232 serial port), and a more versatile programming solution is obtained. Some of the added programming advantages are stand-alone operation when several duplicate EPLDs are needed, increased device testing with checksum, verification, and optional programming of EPROMs and microcontrollers with low cost adaptors.

OPTIONAL PRODUCTS

iPLS: The Intel Programmable Logic Software is available for users who do not require the logic programmer hardware. The product consists of the iPLS diskettes, sample EPLD device, and the *iPLDS User Manual* with slipcase and binder.

iUPLDSKIT09 This kit bundles all the software and hardware needed to develop EPLD's for users that already have an iUP-200A or 201A programmer. The product includes the iPLS software, iUP-GUPI module, GUPI-LOGIC09 adaptor, iPPS programming software for the IBM PC, and manuals.

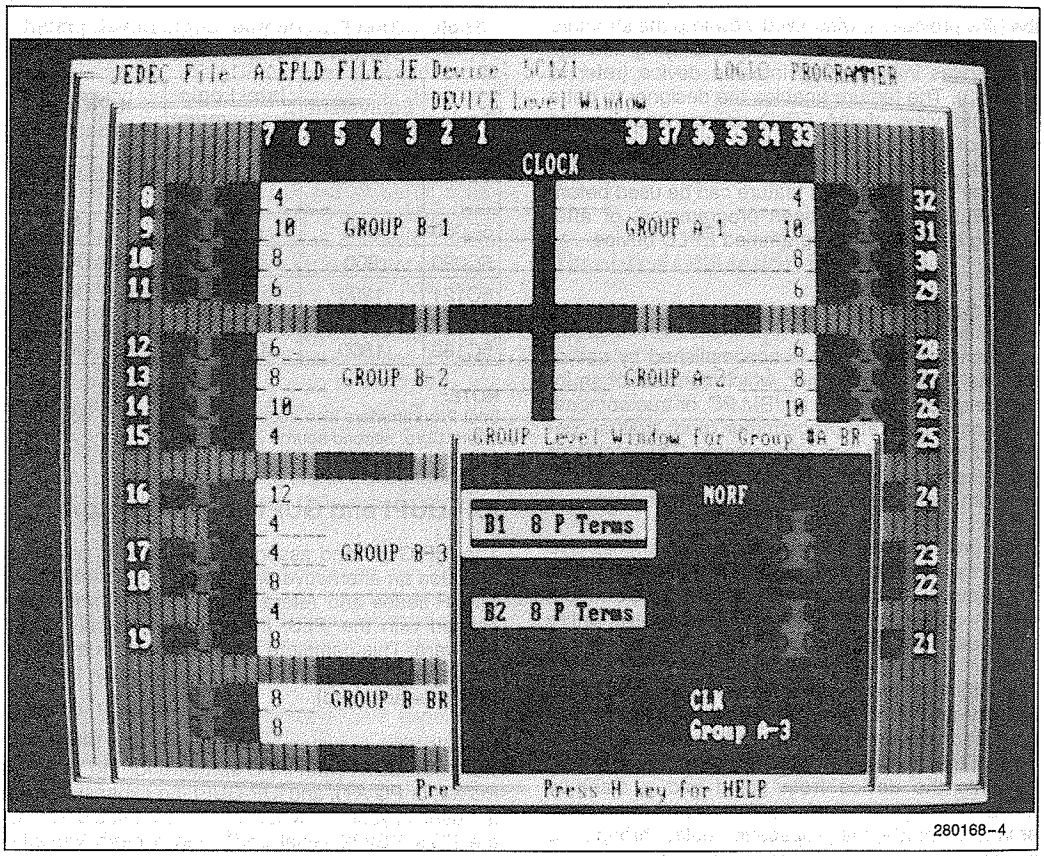


Figure 3. Gate Interconnect Preview Screen Sample

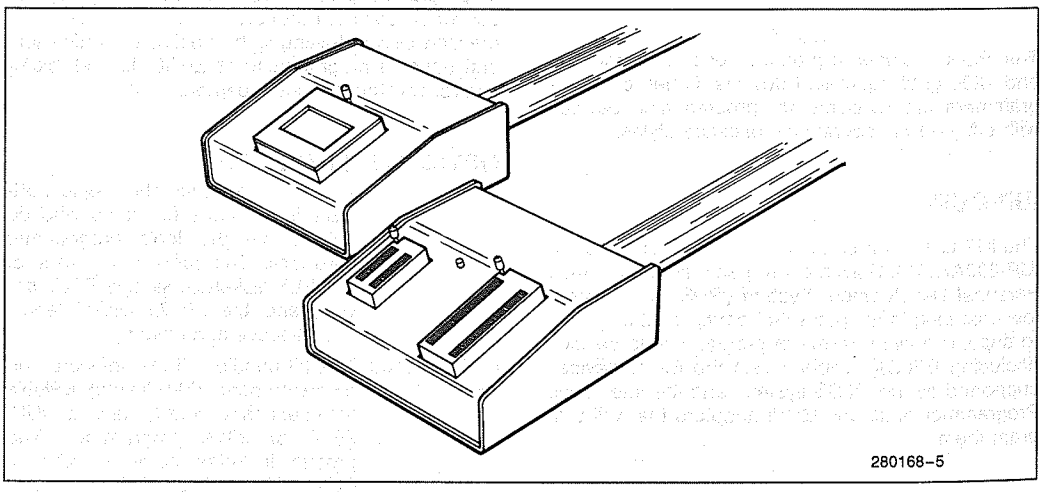


Figure 4. Programmer Pods

Feature	Benefit
• Multiple input formats	Choice of design input methods fits the designer's background, skill level and circumstances
—Schematic capture	Widely used design methodology, provides printout of the schematic
—Logic builder	Very fast to design (no syntax errors associated with text files), easy to use, accepts Boolean equations, lower cost for some users
Boolean algebra	Traditional methodology for programmable logic
—State machine	Support for users designing with state machines
• Self contained, total system	Total, low-cost EPLD design process (development and programming)
• Logic Optimizing Compiler (LOC)	
—Automatic logic minimization and resource allocation	Optimized device utilization (allowing larger equivalent gate counts per device)
—Automatic pin assignment	Design time saved, more efficient design fit within the EPLD device
—Resource utilization report	Automatic documentation (saves time) which provides feedback on the internal implementation of the design
—JEDEC file output	Industry standard, interfaces to third party programmers
• Interactive menu driven software with help files and advanced control features for experts	Very easy to use for the novice, while the expert can quickly access the most sophisticated features
• IBM PC compatible	Minimum host computer investment, widely available, runs on MS-DOS* and PC-DOS
• Sample EPLD devices	Immediate use of the system

iSLIBFNET The FutureNet system library provides the full device library to support both Intel and Altera Corp. EPLD devices, and all necessary software interfacing between the DASH-2 schematic capture package and iPLS. The product consists of the symbol library diskette.

iSLIBPCAD The P-CAD system library provides the full device library to support both Intel and Altera Corp. EPLD devices, and all necessary software interfacing between the PC-CAPS schematic capture package and iPLS. The product consists of the symbol library diskette.

iLP900 The Intel Logic Programmer 900 pod is available for programming the Intel 5C060 and 5C090 logic devices (or equivalent Altera Corp. second-sourced parts).

iLP1800 The Intel Logic Programmer 1800 pod is available for programming the Intel 5C180 logic device (or an equivalent Altera Corp. second-sourced part).

iSTATE The Intel state machine entry software package is available for entering state machine designs by specifying the state variables and state transitions.

SUMMARY

The Intel Programmable Logic Development System is a unique combination of power, versatility, and economics. It enables the logic designer to draw the schematic, check it for accuracy, compile and minimize it, program it into an EPLD device, and then revise the design and reprogram the EPLD device, all at the designer's desk.

SPECIFICATIONS

Required Hardware

The iPLDS software requires an IBM PC XT, PC AT, or other true compatible computer capable of running MS-DOS* version 2.0 or later. The computer must have a 360 KB double-sided, double-density disk drive, a hard disk, and 512 KB of RAM. Additional memory is required for the optional schematic capture programs. A color monitor is recommended, as the color graphics available provide a better representation of the data than a monochrome display.

*MS-DOS is a registered trademark of Microsoft Corporation.

The programmer interface card requires one full-size card slot in the host computer.

Operating Environment

ELECTRICAL CHARACTERISTICS

Interface card and programmer:

Static: 5V @ 300 mA ±50 mA
12V @ 200 mA ±25 mA

Dynamic: 5V @ 300 mA ±50 mA

(programming) 12V @ 250 mA ±25 mA + device current*

NOTE:

*device current = I_{PP} + I_{CC} of the device being programmed

PHYSICAL CHARACTERISTICS

Interface card:
Width: 13.1 in. (33.7 cm)
Height: 4.2 in. (10.8 cm)

Programmer:
Width: 4.8 in. (12.3 cm)
Height: 1.9 in. (4.9 cm)
Depth: 4.8 in. (12.3 cm)

Shipping weight; 7 lbs.

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 10°C to 40°C
Relative Humidity: 8% to 80%

Equipment Supplied

HARDWARE

— Intel Logic Programmer and cable
— Logic programmer interface card
— EPLD device

SOFTWARE

— Intel Programmable Logic Software (iPLS)-master program diskette
— Logic Builder (LB)-design entry diskette
— Logic Optimizing Compiler (LOC) diskette
— Logic Programmer Software (LPS) diskette
— Installation (INSTALL) diskette

DOCUMENTATION:

— *iPLDS User Manual,* order number 166612

ORDERING INFORMATION

Product Order Code	Description
iPLDS	Intel Programmable Logic Development System (hardware, software, sample device, and documentation)
iPLS	Intel Programmable Logic Software (software, sample device, and documentation only)
iSLIBFNET	FutureNet symbol library for use with the DASH-2 schematic capture package
iSLIBPCAD	P-CAD symbol library for use with the PC-CAPS schematic capture package
iLP900	iLP pod for programming Intel 5C060 and 5C090 logic devices
iLP1800	iLP pod for programming Intel 5C180 devices
iSTATE	iPLDS state machine entry software package

NOTE:

The DASH-2 schematic capture program is available from FutureNet Corporation, and the PC-CAPS schematic capture program is available from P-CAD Corporation.